Beautiful Soup库

Beautiful Soup库是能够解析HTML或XML文件的功能库

html\_doc = """<html><head><title>The Dormouse's story</title></head><body><p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;and they lived at the bottom of a well.</p>

<p class="story">...</p>"""

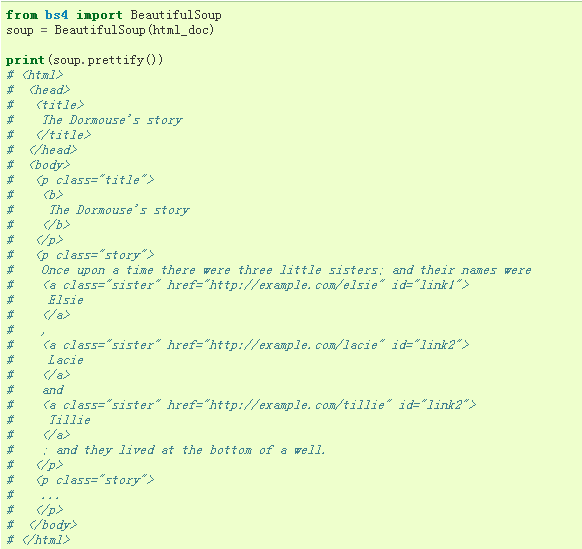
例如每一段HTML代码都是由尖括号组成的“标签树”，Beautiful Soup库能够解析、遍历、维护这个“标签树”

|  |
| --- |
| **from** bs4 **import** BeautifulSoup **import** requests  r **=** requests.get**("http://www.baidu.com")** soup **=** BeautifulSoup**(**r.text, **"html.parser")** print**(**soup.prettify**())**  <!DOCTYPE html>  <!--STATUS OK-->  <html>  <head>  <meta content="text/html;charset=utf-8" http-equiv="content-type">  <meta content="IE=Edge" http-equiv="X-UA-Compatible">  <meta content="always" name="referrer">  <link href="http://s1.bdstatic.com/r/www/cache/bdorz/baidu.min.css" rel="stylesheet" type="text/css">  <title>  ç¾åº¦ä¸ä¸ï¼ä½ å°±ç¥é  </title>  </link>  </meta>  </meta>  </meta>  </head>  <body link="#0000cc">  <div id="wrapper">  <div id="head">  <div class="head\_wrapper">  <div class="s\_form">  <div class="s\_form\_wrapper">  <div id="lg">  <img height="129" hidefocus="true" src="//www.baidu.com/img/bd\_logo1.png" width="270">  </img>  </div>  <form action="//www.baidu.com/s" class="fm" id="form" name="f">  <input name="bdorz\_come" type="hidden" value="1">  <input name="ie" type="hidden" value="utf-8">  <input name="f" type="hidden" value="8">  <input name="rsv\_bp" type="hidden" value="1">  <input name="rsv\_idx" type="hidden" value="1">  <input name="tn" type="hidden" value="baidu">  <span class="bg s\_ipt\_wr">  <input autocomplete="off" autofocus="" class="s\_ipt" id="kw" maxlength="255" name="wd" value=""/>  </span>  <span class="bg s\_btn\_wr">  <input class="bg s\_btn" id="su" type="submit" value="ç¾åº¦ä¸ä¸"/>  </span>  </input>  </input>  </input>  </input>  </input>  </input>  </form>  </div>  </div>  <div id="u1">  <a class="mnav" href="http://news.baidu.com" name="tj\_trnews">  æ°é»  </a>  <a class="mnav" href="http://www.hao123.com" name="tj\_trhao123">  hao123  </a>  <a class="mnav" href="http://map.baidu.com" name="tj\_trmap">  å°å¾  </a>  <a class="mnav" href="http://v.baidu.com" name="tj\_trvideo">  è§é¢  </a>  <a class="mnav" href="http://tieba.baidu.com" name="tj\_trtieba">  è´´å§  </a>  <noscript>  <a class="lb" href="http://www.baidu.com/bdorz/login.gif?login&amp;tpl=mn&amp;u=http%3A%2F%2Fwww.baidu.com%2f%3fbdorz\_come%3d1" name="tj\_login">  ç»å½  </a>  </noscript>  <script>  document.write('<a href="http://www.baidu.com/bdorz/login.gif?login&tpl=mn&u='+ encodeURIComponent(window.location.href+ (window.location.search === "" ? "?" : "&")+ "bdorz\_come=1")+ '" name="tj\_login" class="lb">ç»å½</a>');  </script>  <a class="bri" href="//www.baidu.com/more/" name="tj\_briicon" style="display: block;">  æ´å¤äº§å  </a>  </div>  </div>  </div>  <div id="ftCon">  <div id="ftConw">  <p id="lh">  <a href="http://home.baidu.com">  å³äºç¾åº¦  </a>  <a href="http://ir.baidu.com">  About Baidu  </a>  </p>  <p id="cp">  ©2017 Baidu  <a href="http://www.baidu.com/duty/">  ä½¿ç¨ç¾åº¦åå¿è¯»  </a>  <a class="cp-feedback" href="http://jianyi.baidu.com/">  æè§åé¦  </a>  äº¬ICPè¯030173å·  <img src="//www.baidu.com/img/gs.gif">  </img>  </p>  </div>  </div>  </div>  </body>  </html>  Process finished with exit code 0 |

注意代码 soup = BeautifulSoup(r.text, "html.parser") 是将用requests.get方法获得的服务器返回给我们HTML文件（也就是r.text）转换成BeautifulSoup类，这样我们就可以对这个HTML进行解析了。

可以将HTML文件、标签树、BeautifulSoup类理解为一一对应的关系，转换成soup变量，我们就可以进行一系列操作了，例如如下操作

soup.prettify()就是将HTML文件进行更好的展示的函数，使得每一个标签对（如<p>...</p>）都能对齐，中间即为内容，这样看起来更方便



这是开头文档的prettify()之后的形式

Beautiful Soup将复杂HTML文档转换成一个复杂的树形结构,每个节点都是Python对象。

Tag: <a href="http://ir.baidu.com">

About Baidu

</a>

这个就是一段Tag，name为a

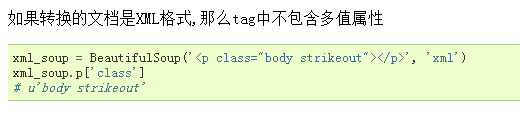


注意第一段代码，如果这段HTML文字中有多个标签b，那么soup.b只返回第一个



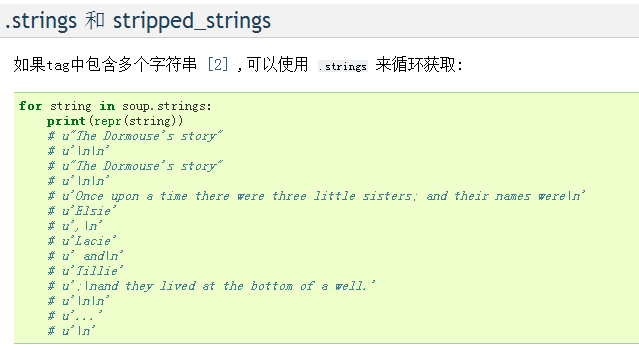
Attributes是多值属性的意思

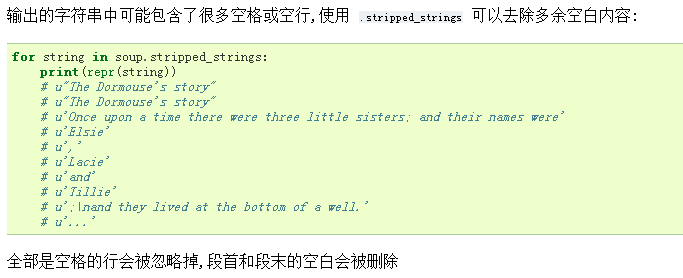






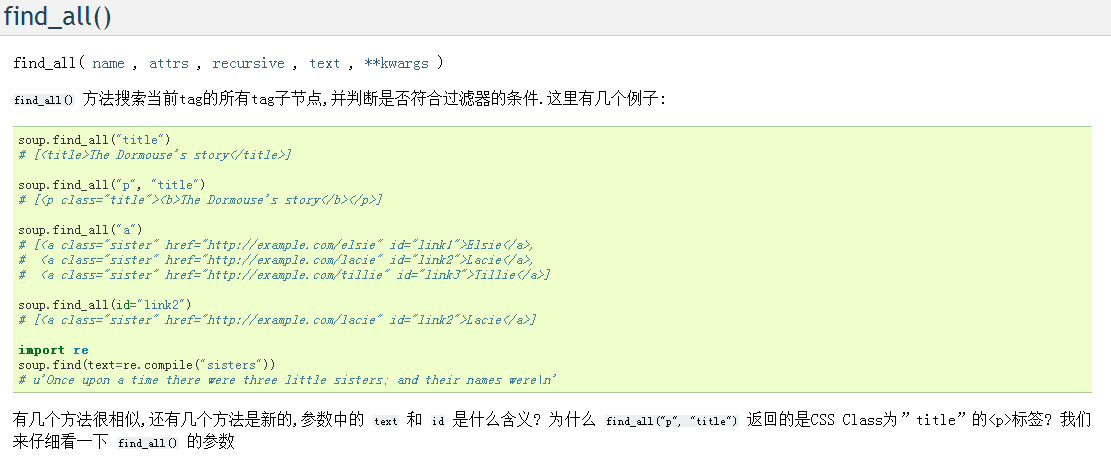


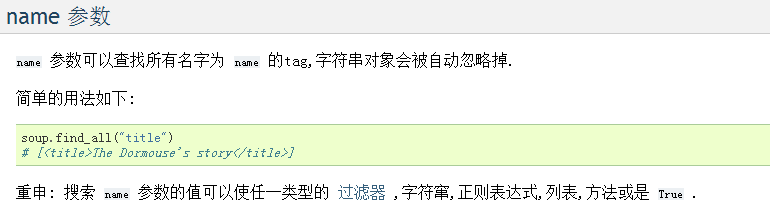


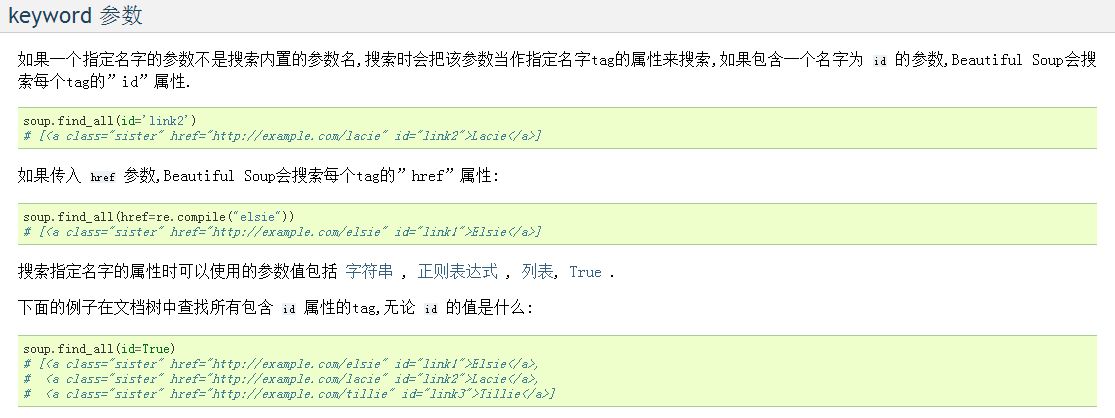


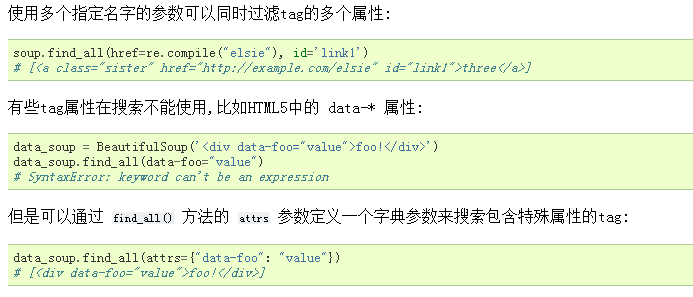














注意find()只返回一个结果，且返回的类型是<class 'bs4.element.Tag'>

find\_all()返回的是一个列表，返回的类型是<class 'bs4.element.ResultSet'>

Tag就可以继续使用find()和find\_all()函数